BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

Compiler Construction (CS F363)
II Semester 2024-25
Compiler Project
Coding Details
(March 15, 2025)

Group Number

22

1. Team Members Names and IDs

ID:	2021B3A70981P	Name:	Anirudh Anand
ID:	2021B3A71738P	Name:	Akshit Phophaliya
ID:	2021B3A70995P	Name:	Dhruv Ravi Krishnan
ID:	2022A7PS1182P	Name:	Arnav Dham
ID:	2022A7PS0187P	Name:	Darsh Rathi
ID:	2022A7TS0154P	Name:	Shaurya Jain

2. Mention the names of the Submitted files :
   1. grammar.txt
   2. coding_details.pdf
   3. lexerDef.h
   4. lexer.h
   5. lexer.c
   6. parserDef.h
   7. parser.h
   8. parser.c
   9. driver.c
   10. makefile
   11. testcase1.txt
   12. testcase2.txt
   13. testcase3.txt
   14. testcase4.txt
   15. testcase5.txt
   16. testcase6.txt

3. Total number of submitted files (including copy the pdf file of this coding details pro forma) : 16 (All files should be in ONE folder named as Group_#)

4. Have you compressed the folder as specified in the submission guidelines? (yes/no) Yes

5. **Lexer Details:**

   [A]. Technique used for pattern matching: Longest match principle

   [B]. Keyword Handling Technique: We have used a hash map to identify keywords

   [C]. Hash function description, if used for keyword handling: Computes a hash value for a given string using an optimized version of the DJB2 algorithm. This function is well-suited for keyword hashing in hash tables, ensuring a good distribution of values while maintaining efficiency.

   [D]. Have you used twin buffer? (yes/ no) Yes

   [E]. Error handling and reporting (yes/No): Yes

   [F]. Describe the errors handled by you Invalid tokens are detected based on our DFA. We also detect scenarios where the parser runs out of elements to parse before the tree is complete as well as when the file contains extra inputs after the end of the source code.

   [G]. Data Structure Description for tokenInfo (in maximum two lines): returnTK stores lexical token details, including its type (Token t), lexeme, line number, and a flag for additional details(like EOF,error etc).

6. **Parser Details:**

[A]. High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):

i. grammar : We stored the grammar as an array of sets of linked lists. Each array index corresponds to the production rules for that particular non-terminal. Each element of the set is linked to the next element in a linked list format.

ii. FIRST and FOLLOW sets : The first and follow sets are stored as arrays of linked list. Each array index corresponds to a non-terminal storing a linked list of terminals belonging to that particular first/ follow set.

iii. parse table: Stored as a 2-D array containing production rules as their elements. A production rule is stored as a custom data structure containing the LHS element, the RHS element and the number of terms/ elements in the RHS.

iv. parse tree: (Describe the node structure also): A custom data structure called TreeNode has been used which contains pointers to the parent TreeNode and its children along with their count. Along with this it also contains information regarding the Node symbol, token lexeme, line number and value in case of real and natural numbers.

v. Any other (specify and describe): The elements (terminals and non-terminals) have been stored in Enums to facilitate easy access as and when required.

[B]. Parse tree

i. Constructed (yes/no): Yes

ii. Printing as per the given format (yes/no): Yes

iii. Describe the order you have adopted for printing the parse tree nodes (in maximum two lines) In-order-traversal has been used to print elements of the parse tree. Panic mode recovery has been implemented to ensure parse tree generation is not completely halted in case of errors.

[C]. Grammar and Computation of First and Follow Sets

i. Data structure for original grammar rules: A set of linked lists corresponding to the given non-terminal (stored as an array of sets of linked lists).

ii. FIRST and FOLLOW sets computation automated (yes /no): Yes

iii. Name the functions (if automated) for computation of First and Follow sets: computeFirst() and computeFollow()

iv. If computed First and Follow sets manually and represented in file/function (name that): NA

[D]. Error Handling

v. Attempted (yes/ no): Yes

vi. Describe the types of errors handled: Invalid symbol recognition, invalid token generation based on the DFA submitted. For parse tree generation errors pertaining to grammatical inaccuracy, extra symbol detection and files containing incomplete source code have been detected.

7. Compilation Details:

[A]. Makefile works (yes/no): Yes

[B]. Code Compiles (yes/ no): Yes

[C]. Mention the .c files that do not compile: NA

[D]. Any specific function that does not compile: NA

[E]. Ensured the compatibility of your code with the specified gcc version (yes/no): yes the code has been tested in an Ubuntu 22.04 environment containing gcc – 11.4.0

8. Driver Details: Does it take care of the options specified earlier(yes/no): Yes

9. Execution

[A]. status (describe in maximum 2 lines): Runs as desired on all testcases.

[B]. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the testcase file name: NA

10. Specify the language features your lexer or parser is not able to handle (in maximum one line): NA

11. Are you availing the lifeline (Yes/No): No

12. Declaration: We, **Anirudh Anand, Akshit Phophaliya, Dhruv Ravi Krishnan, Arnav Dham, Darsh Rathi and Shaurya Jain** declare that we have put our genuine efforts in creating the compiler project code and have submitted the code developed only by us. We have not copied any piece of code from any source. If our code is found plagiarized in any form or degree, we understand that a disciplinary action as per the institute rules will be taken against all of us in our team and we will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

Your names and IDs

Name:	Anirudh Anand	ID:	2021B3A70981P
Name:	Akshit Phophaliya	ID:	2021B3A71738P
Name:	Dhruv Ravi Krishnan	ID:	2021B3A70995P
Name:	Arnav Dham	ID:	2022A7PS1182P
Name:	Darsh Rathi	ID:	2022A7PS0187P
Name:	Shaurya Jain	ID:	2022A7TS0154P

Date: 15/03/24

Not to exceed 3 pages.